

Allen-Bradley Rockwell – ControlLogix/CompactLogix Families

The ControlLogix protocol can be used in two independent ways: with a TagProvider configuration or as a Device > Channel > Node > Point configuration. A TagProvider allows you to connect with data directly, without creating local tags, using the tag names from the remote device. A DeviceChannel allows you to map tags to external data addresses, which provides more flexibility for name conventions and for managing data blocks.

Summary Information

- Communication Driver Name:** ControlLogix
- Implementation DLL:** T.ProtocolDriver.ControlLogix.dll
- Protocol:** ControlLogix (CIP over TCP/IP)
- Interface:** TCPIP
- PLC types supported:** ControlLogix 5000 family, FlexLogic, and CompactLogix
- Manufacturer:** Allen-Bradley / Rockwell
- PC Hardware requirements:** Ethernet board

Channel Configuration

Protocol Options

- Model:** The PLC model can be:
- Others: For all models except 1756-L8X
 - 1756-L8X: For Allen-Bradley model: 1756-L8X, 5069-L330ERS, CompactLogix 5380 and 5069-L310ERS2 5380 GuardLogix.
- MaxStringItemsPerBlock:** Defines the maximum string item count for each block. For example: If there are 12 string items configured and the MaxStringItemsPerBlock is 3, the result will be 4 blocks to read string data.

Node Configuration

Station Configuration

- Stations syntax: <IP>;<Port>;<Slot>
- Where:
- <IP> = The IP address of the slave device in the network
 - <Port> = The TCP port where the slave device is listening (default is 44818)
 - <Slot> = The Slot number where the CPU is connected

Example Node Configuration

Name	Node	PrimaryStation	SecondaryStation	Description
Node1	ControlLogix	192.168.1.101;44818;0		

Points Configuration

Address Column Configuration

The syntax for the ControlLogix communication points is: <Type>:<DeviceTagName>

Where:

- **<Type>** is the data type of a tag in a PLC. The valid type values are:

Type	Read	Write	Size	Range of Value
BOOL	yes	yes	1 bit	0 or 1
SINT	yes	yes	1 byte or 8 bits	-128 to 127
INT	yes	yes	2 bytes or 16 bits	-32768 to 32767
DINT	yes	yes	4 bytes or 32 bits	-2,147,483,648 to 2,147,483,647
REAL	yes	yes	4 bytes or 32 bits IEEE Floating point	-9.99x1037 to 9.99x1037
STRING	yes	yes	n bytes	---

- **<DeviceTagName>** is a Tag Name in a PLC. Possible formats for DeviceTagName:

Format	Syntax	Comments
Standard	<Device Tag Name>	
Array Element	<Device Array Tag name>[dim 1, dim2, dim3]	Dimension Range = 1 to 3
User Defined Type (UDT)	<Device Main Tag Name>.<Type Member>	
User Defined Type (UDT) Array Element	<Device Main Tag Name>.< Type Member>[dim 1, dim2, dim 3]	Dimension Range = 1 to 3

Accessing Bit from SINT, INT, and DINT ControlLogix Datatypes

Method 1: The Modifiers Column

Use the Modifiers column in Device > Points to specify which Bit will be accessed

Example of a Device Point Configuration that will Access Bit 3 from INT

TagName	Node	Address	DataType	AccessType	Modifiers
Integer	Node1	INT:PLC_INTEGER	Native	ReadWrite	Bit = 3

Method 2: The Tag Property

In the Device the whole word into a tag and use the Tag property to access the specify Bit.

Example of a Device Point Configuration that will Access a Whole Word

TagName	Node	Address	DataType	AccessType
Integer	Node1	INT:PLC_INTEGER	Native	ReadWrite

Syntax that will Access a Tag Property in Any Area of a Project

Tag.<TagName>.Bit<Bit Number>

E.g.: Tag.Integer.Bit3

Example Points Configuration

TagName	Node	Address	DataType	AccessType
Digital	Node1	BOOL:PLC_BOOLEAN	Native	ReadWrite
Digital[2]	Node1	BOOL:BOOLEAN_ARRAY[2]	Native	ReadWrite
DigitalUDT	Node1	BOOL:MAINTAG.PLC_BOOLEAN	Native	ReadWrite
DigitalUDT[4]	Node1	BOOL:MAINTAG.BOOLEAN_ARRAY[4]	Native	ReadWrite
Word	Node1	SINT:PLC_SINTEGER	Native	ReadWrite
Word[7]	Node1	SINT:SINTEGER_ARRAY[7]	Native	ReadWrite
WordUDT	Node1	SINT:MAINTAG.PLC_SINTEGER	Native	ReadWrite
WordUDT[8]	Node1	SINT:MAINTAG.SINTEGER_ARRAY[8]	Native	ReadWrite
Integer	Node1	INT:PLC_INTEGER	Native	ReadWrite
Integer[3]	Node1	INT:INTEGER_ARRAY[3]	Native	ReadWrite
IntegerUDT	Node1	INT:MAINTAG.PLC_INTEGER	Native	ReadWrite
IntegerUDT[10]	Node1	INT:MAINTAG.INTEGER_ARRAY[10]	Native	ReadWrite
Double	Node1	DINT:PLC_DINTEGER	Native	ReadWrite
Double[14]	Node1	DINT:DINTEGER_ARRAY[14]	Native	ReadWrite
DoubleUDT	Node1	DINT:MAINTAG.PLC_DINTEGER	Native	ReadWrite
DoubleUDT[12]	Node1	DINT:MAINTAG.DINTEGER_ARRAY[12]	Native	ReadWrite
Real	Node1	REAL:PLC_REAL	Native	ReadWrite
RealArray[5]	Node1	REAL:REAL_ARRAY[5]	Native	ReadWrite
RealUDT	Node1	REAL:MAINTAG.PLC_REAL	Native	ReadWrite
RealUDT[34]	Node1	REAL:MAINTAG.REAL_ARRAY[34]	Native	ReadWrite
Text	Node1	STRING:PLC_STRING	Native	ReadWrite
TextArray[26]	Node1	STRING:STRING_ARRAY[26]	Native	ReadWrite
TextUDT	Node1	STRING:MAINTAG.PLC_STRING	Native	ReadWrite
TextUDT[21]	Node1	STRING:MAINTAG.STRING_ARRAY[21]	Native	ReadWrite

Troubleshoot

The status of the driver's execution can be observed through the diagnostic tools, which are:

- Trace window
- Property Watch
- Module Information

The above tools indicate if the operations have succeeded or have failed. A status of 0 (zero) means communication is successful. Negative values indicate internal driver errors, and positive values indicate protocol error codes.

Error Codes

Error Code	Description	Possible Solution
0	Success	<ul style="list-style-type: none"> • None
-100	Error Sending Message	<ul style="list-style-type: none"> • Turn the PLC on • Plug in the PLC ethernet cable • Check the configured IP Address field in Device > Node • Ping the PLC using the prompt command
-101	Error Sending and Waiting Message	

- 102 . . . -105	Error Creating the TCP/IP connection	
-106	Error Receiving Message	
-112	Timeout Start Message	<ul style="list-style-type: none"> • Turn the PLC on • Plug in the PLC ethernet cable • Ping the PLC using the prompt command • Check the configured IP Address field in Device > Node • Increase the driver timeout field in Device > Channel
-113	Timeout between Treated Characters	
-114	Timeout End Message	
-115	Timeout Connect	
-200	Protocol Error	<ul style="list-style-type: none"> • Check if the PLC model is compatible with the driver documentation • Check the configured Address field in Device > Points
-201	Invalid Protocol	<ul style="list-style-type: none"> • Check if the PLC model is compatible with the driver documentation • Contact technical support
-202	Invalid Station	<ul style="list-style-type: none"> • Check the configured IP Address field in Device > Node • Restart the driver
-204	Invalid Message Sequence	<ul style="list-style-type: none"> • Check if the PLC model is compatible with the driver documentation • Check the configured Address field in Device > Point
> 0	CIP Error	<ul style="list-style-type: none"> • See CIP error codes table

CIP Error Codes

Error Code	Description
1	Connection Failure
2	Connection Failure
3	Value invalid
4	IOI could not be deciphered or tag does not exist
5	Unknown destination
6	Data requested would not fit in response packet
7	Loss of connection
8	Unsupported service
9	Error in data segment or invalid attribute value
10	Attribute list error
11	State already exists
12	Object model conflict
13	Object already exists
14	Attribute not settable
15	Permission denied

16	Device state conflict
17	Reply will not fit
18	Fragment primitive
19	Insufficient command data / parameters specified to execute service
20	Attribute not supported
21	Too much data specified
26	Bridge request too large
27	Bridge response too large
28	Attribute list shortage
29	Invalid attribute list
30	Embedded service error
31	Failure during connection
34	Invalid reply received
37	Key segment error
38	Number of IOI words specified does not match IOI word count
39	Unexpected attribute in list

For this driver, it is very important to enable the TraceWindow messages, as invalid addresses can cause the communication block with the PLC to fail. The TraceWindow tool (when Device is enabled in the settings) will display the first invalid address found on the block.

To have a quick view of the many communication blocks, open the ModuleInformation, use the tree to find ControlLogix, and select the Read Groups. The success and fail communication counters allow you to easily identify any blocks with errors. Then, you can use the TraceWindow to locate the incorrect address.